

REMARKS/ARGUMENTS

This Amendment is in response to the Office Action dated March 10, 2004. Claims 2-3, 5-6, 8-9, 11-12, 14-15, 17-18, and 23-25 are pending. All pending claims are rejected. No pending claims have been amended. No claims have been canceled. Claims 26-28 have been added. Accordingly, claims 2-3, 5-6, 8-9, 11-12, 14-15, 17-18, and 23-28 remain pending in the present application.

Claims 2-3, 5-6, 8-9, 11-12, 14-15, 17-18 and 23-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Haderle et al. (6,185,699) and in view of Watts et al. (6,275,832). The Examiner states:

As per independent claims 23-25, Haderle rendered by the following:
“determining that at least one computer system of the plurality of computer systems has failed” at Fig. 1, col. 5, lines 47-49;
“performing a restart operating on the failed system to free the retained locks using only shared processor resources determined to be necessary for performing the restart operation” at Fig. 1, col. 5, lines 53-57.

Haderle does not teach specifically retaining locks at the time of restarting the system after failure. However, Watts teaches the following:
“retaining a plurality of locks held by the failed system in response to the failure” at Fig. 3, col. 7, line 65 to col. 8, line 14.

Thus, it would have been obvious to one ordinary skilled in the art at the time of the invention to incorporate computer-programming instructions to convert nonstandard database record to a standard database record. Haderle and Watts are combined as they teach recovery techniques from database failure and to retain of locks during database restarting time. In order to undo a transaction backout from system failure retained lock information is useful...

In the section “Response to Arguments”, the Examiner states:

...In response to the Applicants’ argument, the prior art by Haderle and Watts still teaches all claims its limitations. For information, Haderle teaches restarting the system automatically in response to the failure, or waits for a user command to restart, the recovery mechanism makes an analysis pass through the log from the last check-point forward (at Fig. 1, col. 5, lines 54-60). Whereas Watts teaches retaining locks to recover the system from failure (at Fig. 3, col. 7, line 65 to col. 8, line 14).

Applicant respectfully disagrees. The present invention, as recited in independent claims

23, 24, and 25, allows the restarting of the failed computer system using only the shared processor resources that are necessary for recovering the data protected by the plurality of locks held by the failed computer system at the time of failure. Resources that do not facilitate the recovery of the data are not used during the restart. Such resources include allowing the failed computer system to accept new work. (New claims 26-28 and Specification p. 4, line 21 – p. 5, line 3)

Haderle discloses a method and apparatus to provide DBMS restart recovery that allows transactions to access data that does not have restart recovery work pending. Haderle states:

Regardless of the embodiment, the invention allows full recovery to be completed concurrent to the processing of new transactions requiring access to the database. An amount of restart recovery processing may be postponed until after the DBMS has begun accepting new work requests.” (Abstract)

Thus, in contrast to the present invention, Haderle allows resources other than that necessary for the recovery of the data that have restart recovery work pending to be used during the restart of the failed computer system. More specifically, during the restart, Haderle allows the failed computer system to accept new work. According to the present invention, resources to allow the failed computer system to accept new work is not used during the restart since this resource is not necessary for recovery the data protected by the retained locks. This allows a significant reduction in the CPU resources and storage requirements for performing the restart, and the restart can be performed significantly faster. (Specification p. 4, line 21 – p. 5, line 3) Haderle cannot provide such advantages.

Therefore, even if Watts teaches the limitation as argued by the examiner, Haderle in view of Watts still does not teach or suggest restarting the at least one computer system using only shared processor resources determined to be necessary for performing the restart operation, in combination with the other recited elements in independent claims 23, 24, and 25. Haderle in view of Watts further does not teach or suggest that the necessary shared resources do not include enabling the

failed system to accept new work, as recited in newly added dependent claims 26-28.

The examiner's arguments fail to point out how Haderle in view of Watts teaches the limitation of performing a restart operation on the failed system to free the retained locks "using only shared processor resources determined to be necessary for performing the restart operation". Nowhere in the examiner's arguments is this limitation addressed. Thus, the examiner has failed to set forth a prima facie case of non-obviousness under 35 U.S.C. 103(a).

Therefore, for the above identified reasons, the present invention as recited in independent claims 23-25 is neither taught nor suggested by Haderle in view of Watts. Applicant further submits that claims 2-3, 5-6, 8-9, 11-12, 14-15, 17-18, and 26-28 are also allowable because they depend on the above allowable base claims.

In view of the foregoing, Applicant submits that claims 2-3, 5-6, 8-9, 11-12, 14-15, 17-18, and 23-28 are patentable over the cited references. Applicant, therefore, respectfully requests reconsideration and allowance of the claims as now presented.

Applicants' attorney believes this application in condition for allowance. Should any unresolved issues remain, Examiner is invited to call Applicants' attorney at the telephone number indicated below.

Respectfully submitted,
SAWYER LAW GROUP LLP

May 24, 2004

Date


Michele Liu
Attorney for Applicant(s)
Reg. No. 44,875
(650) 493-4540